Joint Optimization of NCL PUF Using Frequency-based Analysis and Evolutionary Algorithm

Rabin Yu Acharya and Domenic Forte
Department of Electrical and Computer Engineering, University of Florida, Gainesville, FL, USA
Email: rabin.acharya@ufl.edu and dforte@ece.ufl.edu

***Abstract*—Physically unclonable functions (PUFs) are hardware security primitives which can be used for hardware authentication and cryptographic key generation. The design of PUFs involves configuring the design of existing cells within an integrated circuit (IC) for PUF operation without impacting the normal circuit operation. This makes the design of PUF circuit very challenging especially for analog circuits as they have higher number of design specifications to meet. Thus, the design of PUFs has been explored mainly for digital circuits even though analog ICs are one of the most highly counterfeited circuit types. In this paper, we present a clear and straightforward design methodology that includes automated frequency-based analysis and evolutionary algorithm-based optimization to design a robust and reliable PUF circuit suitable for both analog and digital circuits. Specifically, we present the design of null conventional logic gate based (NCL) PUF that exploits its startup characteristics as a source of entropy. Our previous work explored a delay matching based optimization for transistors of the NCL PUF which was able to obtain a highly unique PUF with fair reliability. In this work, we are able to obtain a more robust PUF circuit with higher reliability across a wide range of temperature ($0^{\circ}C$-$120^{\circ}C$) and supply voltage variations (of up to $\pm 10\%$). We also compare different evolutionary algorithm based techniques to demonstrate the effectiveness of our proposed methodology.**

***Index Terms*—Evolutionary algorithm, Frequency-based analysis, PUF, NCL, AI, Optimization.**

I. Introduction

Physically unclonable functions (PUFs) are a type of hardware security primitive which can be used to create an intrinsic hardware authentication mechanism or generate unique keys for cryptographic schemes. In the last decade or so, they have emerged as the security primitive of choice because of their ability to exploit uncontrollable randomness or *process variation* present in every hardware device during integrated circuit (IC) manufacturing. This capability makes them a low overhead, volatile security primitive resilient against physical (both invasive and non-invasive) attacks, side-channel attacks, and software-based attacks [1], [2]. They have specifically gained traction in the smartcard industry where PUFs are used as a *silicon fingerprint* to create cryptographic keys that are unique to each smartcard [3]. Over forty different PUFs have been proposed to this day most of which are only applicable for digital circuits [4]. A few PUFs are applicable and targeted for analog ICs namely the metal resistance PUF [5], [6],oxide rupture PUF [7], analog electronic PUF [8], [9], [10], Via PUF [11], and our previous work the NCL ARES PUF [12].

For a circuit structure to function as a PUF, we require a source of entropy or intrinsic randomness that can provide random yet reliable output bits which are different from device-to-device only because of manufacturing or process variations. The PUFs described above use different sources of entropy or PUF circuit architectures. The metal resistance PUFs described in [5], [6] exploits the random mismatches in resistance of metal vias across ICs. In oxide rupture PUF [7], a stress voltage is applied intentionally in transistor pairs such that the output bit is determined through the transistor whose oxide breaks first. Analog PUFs leverage threshold voltage mismatch between transistors in cascode current mirror [8], mismatch between pairs of analog circuits proportional-to-absolute-temperature (PTAT) in [9], and mismatch between transistors in current-steering digital-to-analog converter (DAC) and ring voltage-controlled oscillator (VCO) to create two different entropy sources in [10]. Furthermore, the asynchronous reset (ARES) PUF in [12] utilizes the startup characteristics of null conventional logic (NCL) gates to generate output as a result of the metastability induced through process variations. In all of these PUFs depending on the application, different cells within the IC are preselected during the design step for PUF operation. This has become challenging over the years as the size of ICs have shrunk drastically and that there are only limited number of input/output (I/O) pins in an analog IC. Furthermore, its difficult to meet both analog and PUF specifications as analog circuits already have an higher number of design specifications.

As mentioned above, one of the major challenges in designing a PUF circuit is to meet the PUF specifications such as uniformity and reliability. Uniformity is defined as the distribution of 1s and 0s in the output array of the PUF. An even distribution guarantees a strong security as it becomes difficult to replicate or to predict such output. Similarly, reliability is defined as the capability of a PUF circuit in repeating same output values despite changes in environmental conditions such as the supply voltage and the operating temperature [13]. In case of the most popular metastability-driven PUF (SRAM PUF), the static noise margin (PSNM) metric is used to determine the maximum noise startup value that the cell can tolerate before its value flips. This essentially expresses how reproducible and reliable the output of a SRAM PUF is [14]. However, PSNM is only applicable if there are identical structures within the PUF circuit as in SRAM. Our previous work, the design of ARES PUF, uses delay based metrics instead to improve PUF quality in the face of asymmetric NCL gates. Delay based analysis balances the strength of the two inverters present inside the NCL cell such that only the process variation affects the output. However to use this delay-based

metric, the NCL structure is modified such that each inverter inside the cell can be analyzed individually [12]. In this paper, we take the same NCL structure to design an asynchronous PUF suitable for both analog and digital circuits. However, our design methodology does not involve any modification to the original circuit and uses a frequency analysis based method instead of the delay-based analysis technique to size the transistors of the NCL architecture. More specifically, our contributions in this paper are summarized as follows:

- A design methodology using frequency-based analysis that exploits the startup characteristics of NCL gates.
- An automated sizing method based on widely popular evolutionary algorithms such as genetic algorithm (GA) and NSGA-II [15] that uses the fitness function and stopping criteria based on the frequency-based analysis to optimize the NCL netlist for uniformity and reliability.
- Simulation in HSPICE for the typical Threshold 2 of 2 (TH22) NCL PUF with state-of-the-art commercial 65 nm technology node that demonstrates better and robust uniformity and reliability results compared to previous technique [12]. This approach can easily be extended to other NCL gates such as TH33, TH44, TH23, etc. which will be the focus of our future work.

The remainder of the paper is divided as follows, Section II provides background and preliminary information. Section III describes the proposed design methodology. Section IV contains simulation results, and Section V concludes the paper with possible future works.

II. Preliminaries

A. PUF Preliminaries

Silicon PUFs exploit the intrinsic device or manufacturing variations of passive and active elements in an IC to create a device-specific fingerprint. The output of a PUF is recorded in the form of challenge-response pairs (CRPs) which are generated by physically querying the PUF and computing its response. Depending on the number of CRPs, PUFs can be categorized either as (1) Strong PUFs which have an exponential or a large number of CRPs and are generally diffculy for the attacker to create a model to clone and attack the PUF or as (2) Weak PUFs which have a small number of CRPs. Strong PUFs are generally used as an authentication mechanism while Weak PUFs are most suitable for cryptographic key generation.

B. NCL and TH22 Gate Structure

NCL is a quasi delay-insensitive asynchronous logic first introduced by Fant et. al in 1994 [16]. This system of logic gates are symbolically complete and do not need a clock to synchronize their inputs or outputs. NCL uses dual-rail logic which results in three valid states for a gate with two inputs A and B: NULL ($A = 0$, $B = 0$), DATA0 ($A = 1$, $B = 0$) and DATA1 ($A = 0$, $B = 1$). These two signals are mutually exclusive meaning that the last state where both rails are equal to 1 is invalid. NCL gates can only be discrete threshold gates and thus are typically referred to as Threshold (TH) M-of-N

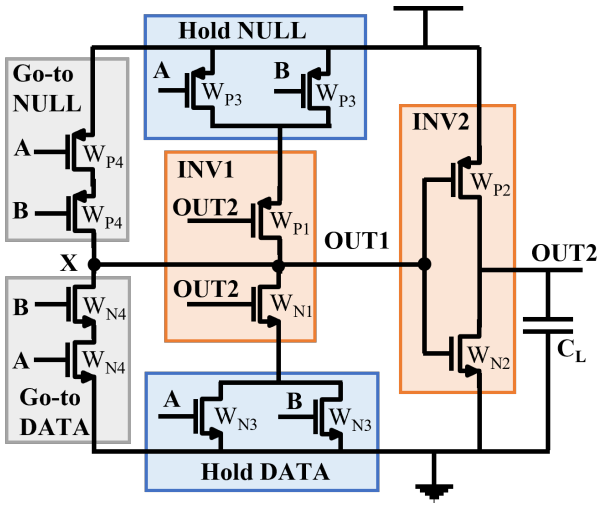

Figure 1: Transistor level diagram of NCL TH22 PUF.

gates. There are a total of 27 different threshold gates that constitute both combinational logic and storage elements with four or fewer number of inputs. These gates have two distinct characteristics: 1) *threshold* which implies that the output transitions from NULL to DATA only if at least M-of-N input conditions become DATA and vice-versa; and 2) *hysteresis* which refers to the ability of the logic gate to retain its output value once a specific number of the input signals are established.

Fig. 1 shows a typical structure of a NCL TH22 gate and TH22 PUF. It consists of four logic blocks, namely the Go-to-NULL, Go-to-DATA, Hold-NULL, and Hold-DATA where NULL and DATA comprise of the three valid states as discussed above. As the name suggests, the Go-to blocks help in switching the state of the output logic from 0 to 1 and vice-versa while the Hold blocks help in retaining the state of the output logic before or after the threshold is met. In this regards, Go-to blocks are only activated when all of the inputs within the block have the same logic value while the hold blocks are activated until at least one of the input has a different logic value compared to the others in the block. In addition to these four blocks, the NCL structure also consists of INV1 and INV2 as shown in the orange blocks of Fig. 1. The transistors in INV1 are referred to as *holding transistors* as they are essential in holding the output logic value while the transistors in INV2 are referred to as *switching transistors* since they help in switching the output logic value.

To use the NCL structure as a source of entropy or PUF, we start the device in a metastable condition meaning that neither the Go-to-NULL and Go-to-DATA blocks are activated. For this purpose, we start the device with the inputs having different value from each other (i.e., $AB = 10$, $AB = 01$). Because of this, the node X in Fig. 1 is floating and the final output of the circuit is completely dependent on the process variations of transistors in the inverters. We leverage this to design a PUF circuit by making sure that the inverters are equal in strength and only manufacturing or process variation tilts the strength in either inverter's favor.

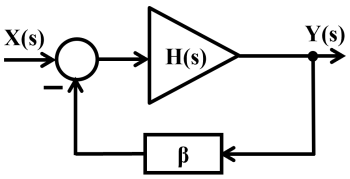

Figure 2: A typical negative-feedback system.

C. Frequency analysis of a feedback system

Fig. 2 shows a typical negative-feedback circuit with an open-loop gain of $|H(s)|$ and feedback of β. The input and output of the system are represented as $X(s)$ and $Y(s)$ respectively. The closed-loop transfer function of this system $T(s)$, the subsequent gain G and phase ϕ is given as follows.

$$T(s) = \frac{Y(s)}{X(s)} = \frac{H(s)}{1 + \beta H(s)}$$

$$Gain, G = |T(s)|_{s=j\omega} = \frac{1}{\sqrt{(Re(T))^2 + \omega^2 (Im(T))^2}}$$

$$Phase, \phi = \angle\, T(s)|_{s=j\omega} = -tan^{-1}\left(\frac{\omega Im(T)}{Re(T)}\right)$$

Based on the above equation of $T(s)$, when $\beta H(s) = -1$ then the closed-loop gain goes to infinity meaning that the system can amplify its own noise until it oscillates. This means that for the circuit to oscillate at a given frequency ω, $\beta|H(s)|_{s=j\omega}$ and ϕ must be at least 1 and 180° respectively. These conditions are also referred to as the Barkhausen's criteria [17]. For the NCL TH22 system that we analyse in this paper (Fig. 1), the feedback β is unity and $H(s)$ is the combined gain of the two inverters. Based on the discussion above, we want the combined gain to be greater than or equal to 1 and the phase shift to be less than 180°.

III. Proposed Methodology

The design of NCL TH22 PUF involves using the structure described in Section II-B and shown in Fig. 1 and then optimizing the value of the components (transistors and the output capacitor) automatically to obtain a reliable source of entropy needed for a successful PUF operation. The circuit is the same as described in our previous work [12]; however, the design methodology is completely different as discussed below. In our previous work, we used a delay based metric to obtain the appropriate sizes and values for the transistors in the NCL circuit. More specifically, the transistor sizes are obtained such that the delay of the rise and fall paths of the inverters (INV1 and INV2) in Fig. 1 are equal in strength. For this purpose, the feedback loop (OUT2 line that goes from output of INV2 to the input of INV1) is disconnected, a clock is then inserted at the input of INV1 and the corresponding rise and fall times are recorded at OUT1. Similarly, OUT1 line is disconnected, a clock is then inserted and the corresponding rise/ fall times are recorded at OUT2. Then using either linear programming or genetic algorithm, the transistors of the NCL circuit are sized such that the recorded rise and fall times for INV1 and INV2 are almost equal.

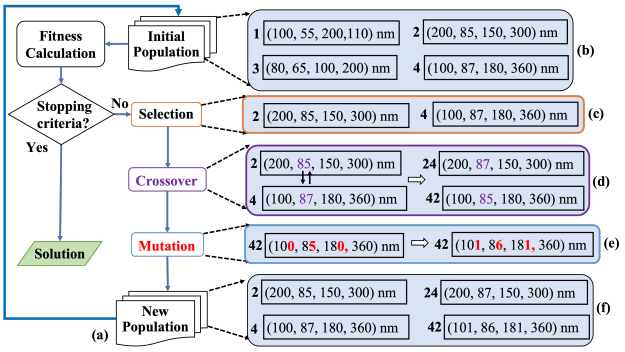

Figure 3: (a) General flow diagram of GA; Example showing (b) population initialization; (c) selection of fit members based on the fitness value; (d) crossover between fit members to create new members; (e) mutation of genes in certain members to create new members; (f) the new population after selection, crossover, and mutation.

In this paper, we instead use a frequency analysis based metric to help size the transistors of the NCL PUF circuit. For this purpose, we do not make any modification to the circuit nor disconnect any wires. Instead we only insert small signal AC source at the inputs of the circuit to be able to obtain a frequency response of the circuit.

A. Genetic Algorithm (GA) Basics

GA is a type of an evolutionary algorithm which is used to generate highly fit solutions to optimization and search problems. Inspired by Charles Darwin's theory of natural section, GA relies on three operators – selection, crossover, and mutation – to generate the "fittest" individual or the best solution to a given problem. These three operators are applied to a population of *genomes* or potential solutions as shown in Fig. 3. As an example, Fig. 3b shows the initial population of 4 genomes where each genome consists of *genes* or width values for transistors in a particular netlist. Then based on the value calculated using the fitness function, fit individuals or best genomes are selected as shown in Fig. 3c. Using these fit individuals, new genomes are created by crossing over some genes or width values between genomes as shown in Fig. 3d. New genomes are also created by mutating or changing values of some genes as seen in Fig. 3e. The new population as a result of these three operators are shown in Fig. 3f. This process of evolution or application of evolutionary operators continues until a stopping criteria is satisfied. The stopping criteria can either be an optimal fitness value or a number of evolutionary runs. In any case, the GA returns the most fit genome that solves the problem at hand.

1) Advantages of GA: GAs are used in various applications that involve solving complex problems such as machine learning, automatic programming, data modeling, and even NP-hard problems (i.e., problems that are non-deterministic and take longer than polynomial time to solve). They are proven to be way more effective to find a global optimum compared to the traditional optimization methods like hill-climbing and gradient-ascent based methods, and exhaustive random search methods [18]. Unlike these methods, GAs are not random

and they do not require extra information like gradients or derivatives. As described above, the only mechanism that guides them is the fitness function along with the stopping criteria. This allows them to function when the search space is noisy, nonlinear, and where derivatives do not exist. However, the fitness function and stopping criteria must be carefully devised to solve any given problem.

2) Formulation of the fitness function: The GA described in Fig. 3 shows a basic evolutionary algorithm that can handle single objective or fitness function smoothly. When we specify multiple objectives, then depending on the problem, the algorithm above might not operate as intended. Instead, a unified fitness function with weights to different objectives must be devised or some advanced algorithms that can handle multiple objectives must be used. Indeed, we explore both of these techniques in this paper to design a NCL-based PUF which is *both* uniform and reliable across different challenges and ICs. The unified fitness function is used in conjunction with the algorithm described in Fig. 3. The unified fitness function is shown in Equation 1 below where M represents the total number of objectives, w_m represents the weight for each objective m and f_m represents a specific objective function.

$$F(x) = \sum_{m=1}^{M} w_m f_m(x) \tag{1}$$

Similarly, regarding the advanced algorithm, we use Non-dominated Sorting Genetic Algorithm (NSGA-II) which is one of the most popular genetic algorithm to tackle multiple objective-based optimization problems [15].

B. NSGA-II

NSGA-II is similar to the algorithm described in Fig. 3. However, the algorithm consists of two additional operators and mechanisms designed specifically for multi-objective optimization: *non-dominated sorting* and *crowding distance*, cf. [15]. Non-dominated sorting is a mechanism where the population of genomes or potential solutions are sorted and partitioned into fronts according to the ascending level of non-domination. The level of non-domination is determined based on how well a specific solution provides a suitable compromise between all objectives without degrading any of them. The non-dominated set of solutions of the entire search space is also referred to as globally *Pareto-optimal set* [19]. Similarly, *crowding distance* is the new parameter calculated by the algorithm to measure the distance between each genome and its neighbors. A large average crowding distance is desired as it results in better diversity and helps in evolving a wide-range of potential solutions potentially leading to quicker convergence. The selection process in this algorithm is based on both fitness function and crowding distance.

C. Methodology to Design NCL TH22 PUF

The design procedure involves evolving the widths of all the transistors in the NCL circuit (Fig. 1) such that the PUF is uniform and reliable across both the challenges (Challenge 1: $AB = 10$, and Challenge 2: $AB = 01$). The widths of the NMOS transistors are referred to as $\boldsymbol{W_N}$ where $\boldsymbol{W_N} = (W_{N1}, W_{N2}, W_{N3}, W_{N4})$ represents the widths of the NMOS transistors of INV1, INV2, Hold DATA, and Goto DATA networks respectively. The widths of the PMOS transistors, $\boldsymbol{W_P}$, can also be evolved but for the sake of simplicity we will use $\boldsymbol{W_P} = 2.5 \times \boldsymbol{W_N}$. Note that the factor of 2.5 is based on the fact that the mobility of electrons is at least twice as high as compared to that of the holes. The detailed design methodology using the GA algorithm described in Section III-A is as follows.

1) For the NCL circuit (Fig. 1) with a given set of $\boldsymbol{W_N}$ and $\boldsymbol{W_P}$, a small signal AC source is inserted at the inputs A and B. The input challenges or the DC values of A and B are set to 1 and 0 and vice-versa representing Challenges 1 and 2, respectively. One thing to note is that since we keep the values of the transistors within a block equal (for example, the transistors in Hold DATA block have same width values), the circuit is not that different across the two challenges.
2) The corresponding frequency response (Gain and Phase margin) is then recorded at OUT1 and OUT2. We require the Gain (G) of the inverters to be equal to each other to ensure that both of the inverters are equal in strength. Similarly, the combined phase margin of the inverters (ϕ_m) is expected to be between 90 and 180 degrees to ensure that the circuit does not oscillate based on the Barkhausen's criteria [17]. For simplicity, we set the criteria of the Gain of the inverters to be equal to 1 or $0dB$.
3) We then use the weighted sum technique as shown in Equation 1 to devise a unified *fitness function* for our evolutionary algorithm such that $M = 2$ and the objective functions are the Gain of the two inverters. We set the value of the weights (w_m) to be equal to each other i.e. 0.5. The resulting equation is shown below. The algorithm as shown in Fig. 3 continues until the fitness value satisfies the criteria described above. This fitness function ensures that the output of the PUF is *uniform* and is only determined by the process variation.
$$F = (0.5 \times G_{INV1}) + (0.5 \times G_{INV2}) \tag{2}$$
4) Once the fitness value is satisfied, we are then ready to stop the algorithm as part of our stopping criteria. In addition to checking if the fitness value is satisfied, we can also add an additional criteria to ensure that the difference of currents (I_{diff}) through INV1 and INV2 is as minimal as possible. This *modified stopping criteria* makes sure that the output of the circuit is affected minimally with changing environmental conditions such that the PUF is *reliable*.

The above methodology uses the evolutionary algorithm described in Section III-A to evolve the transistor sizes of the NCL PUF circuit until it produces uniform and reliable outputs. We also use the NSGA-II algorithm described in Section III-B in a similar fashion as explained above where the multi-objective function comprises of the two fitness functions,

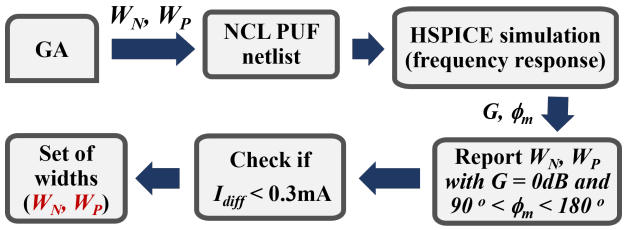

Figure 4: Simulation procedure used to obtain a set of widths for the NCL PUF.

namely the Gain of the two inverters INV1 and INV2.

IV. Simulation Results and Analysis

A. Simulation Setup

The netlist shown in Fig. 1 is simulated in 65 nm technology node using HSPICE. The creation of netlist, subsequent HSPICE simulations, and the evolution of the transistor sizes are all done in an automated fashion within the Python environment as shown in Fig. 4. We implement our own version of GA as shown in Fig. 3, while we use the code developed by Pymoo [20] to implement the NSGA-II algorithm. The parameters related to the GA algorithm such as the number of iterations, crossover rate, mutation rate, number of genomes in a population are set to the standard values of 100, 0.35, 0.1, and 10 respectively.

B. Simulation Procedure

The design methodology described in Section III is used to get a list of transistor sizes such that the subsequent NCL TH22 PUF circuit produces uniform and reliable outputs. For this purpose, we insert small AC signals of $1mV$ and $10kHz$ frequency at the inputs and set the supply voltage V_{DD} at $1V$. The value of the output load capacitor of the NCL PUF circuit, C_L, is set dynamically based on the capacitances seen by the inputs A and B. The corresponding gain and phase margin of the inverters are measured across the frequency range of $0Hz$ to $10kHz$. Then with the evolutionary algorithms, a netlist with a specific set of $\boldsymbol{W_N}$ and $\boldsymbol{W_P}$ that satisfies the fitness criteria of the combined gain, $G = 0dB$ and phase margin of $90° < \phi_m < 180°$ is found. The algorithms run until we wish to continue it to optimize for reliability using the modified stopping criteria discussed in Section III which is set to be $0.3mA$. If the difference in current through the inverters (I_{diff}) is less than $0.3mA$, then the changes in environmental conditions such as supply voltage and temperature will affect the output value of the PUF minimally as will be shown in the results below.

After a netlist is returned using the evolutionary algorithm, we set the supply voltage V_{DD} as a piece-wise linear (PWL) voltage function that increases linearly from $0V$ to $1V$ in a very short span of time ($10fs$) and stabilizes at $1V$ for some time ($1ns$). This simulates the startup behavior as discussed in Section II-B. We then run 1000 Monte Carlo (MC) simulations to mimic different instances of the NCL PUF chip. The MC simulation constitutes of a 5% intra-die process variation and 15% inter-die process variation. This variation is obtained from the MC library provided by the particular process foundry. The MC simulations carried out are different transient simulations which help us obtain a transient output at node Z. This transient output is then quantified to either a logic 0 or a logic 1 based on the threshold value of 0.5. The maximum transient value after the output is settled, which is usually after $10fs$ in our case, is used to get the transient output. The subsequent 1000 outputs are then analyzed to calculate the distribution of logic 1s and 0s.

TABLE I: Comparison of the uniformity results using different optimization methods and metrics.

Optimization method	Uniformity
GA using delay-based metric	51.3 %
GA using frequency-based metric	50.6 %
NSGA-II using frequency-based metric	50.8 %

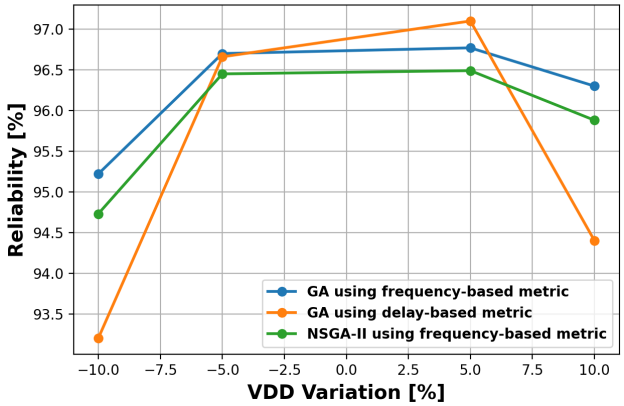

Figure 5: Reliability results of the NCL TH22 PUF across different supply voltage variations.

C. Uniformity Results

Based on the 1000 outputs obtained from the transient MC simulations, the percentage distribution of zeros or ones ($0s$ in our case) is used to calculate the Uniformity value. We perform the procedure mentioned above using the GA algorithm described in Section III-A and compare it with the results when using a similar procedure but with a delay-based metric as described in [12]. We also compare these results when using the same procedure mentioned above but using the NSGA-II algorithm instead. These results and comparison are summarized in Table I where the uniformity results are very comparable to each other. More specifically, the results are seen to be better when using the frequency-based metric.

D. Reliability Results

Using the same netlist that produced the 1000 transient outputs described above, we test the reliability of that circuit by running the transient MC simulations across different supply voltages and temperatures. In this regard, we consider the supply voltage variations of up to $\pm 10\%$ and the temperatures ranging from $0°C$ to $120°C$. Then, we check and see how each voltage bit changes as a result of the change in environmental conditions (supply voltage and temperature). The percentage number of bits that do not flip its original logical state is

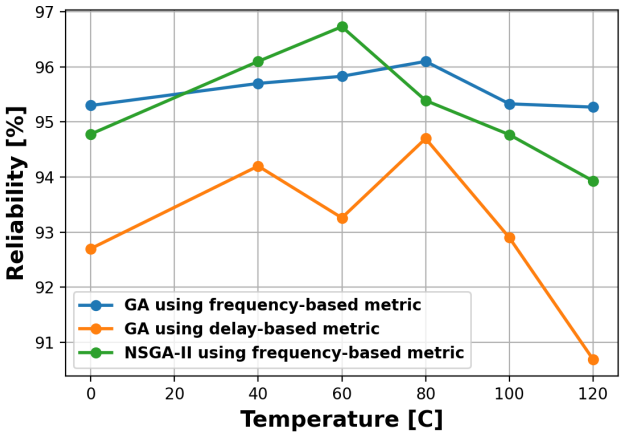

Figure 6: Reliability results of the NCL TH22 PUF across different temperatures.

used as the reliability value. For comparison, we also compile these results using the GA with the delay-based metric and the NSGA-II algorithm with the frequency-based metric. The reliability results across voltage variations and temperature variations are summarized in Fig. 5 and Fig. 6, respectively. The frequency-based analysis produces better reliability with respect to both voltage and temperature compared to delay-based analysis. Another thing to note is that the results for the frequency-based metric were obtained using the evolutionary algorithm with the modified fitness criteria. Without this criteria, it is not guaranteed whether the netlist produced by the algorithm will be reliable or not. In fact, the results for the delay-based metric were obtained after a few number of GA runs and manual checks as the initially generated netlists produced very unreliable output bits.

Furthermore, looking at the reliability results in Figures 5 and 6, it might not be clear as to why a sophisticated multi-objective optimization algorithm such as NSGA-II needs to be implemented since the results are very comparable. One reason is to compare the performance of our version of the GA to the state-of-the-art evolutionary algorithm. The second reason is because NSGA-II is very fast (around 3 times) in converging to a uniform and reliable PUF when compared to the simple GA that we devised. It is the case that NSGA-II can handle multi-objective optimization functions really well; however, we hope that it can successfully help design complicated NCL circuits such as the TH44, TH32, etc. with similar ease which will be the focus of our future work.

V. Conclusion and Future Work

In this paper, we demonstrated a novel design methodology to design a uniform and reliable NCL TH22 PUF using the frequency analysis based metric and the evolutionary algorithm. We also introduced a modified stopping criteria to help ensure that the PUF netlist generated by the algorithm produces highly reliable bits. In this regards, we generated a PUF netlist which is 50.6% uniform and on average 96.2% and 95.6% reliable across supply voltage variations of $\pm 10\%$ and temperatures ranging from $0°C$-$120°C$, respectively. In the future, we plan to fabricate these netlists in Silicon. We will also extend the design methodology of using both of the evolutionary algorithms discussed to design more complex NCL circuits such as the TH44, TH32, etc.

References

[1] N. Beckmann and M. Potkonjak, "Hardware-based public-key cryptography with public physically unclonable functions," in *International Workshop on Information Hiding*, pp. 206–220, Springer, 2009.

[2] B. Gassend, D. Clarke, M. Van Dijk, and S. Devadas, "Silicon physical random functions," in *Proceedings of the 9th ACM Conference on Computer and Communications Security*, pp. 148–160, 2002.

[3] V. van der Leest, R. Maes, G.-J. Schrijen, and P. Tuyls, "Hardware intrinsic security to protect value in the mobile market," in *ISSE 2014 Securing Electronic Business Processes*, pp. 188–198, Springer, 2014.

[4] T. McGrath, I. E. Bagci, Z. M. Wang, U. Roedig, and R. J. Young, "A puf taxonomy," *Applied Physics Reviews*, vol. 6, no. 1, p. 011303, 2019.

[5] R. Helinski, D. Acharyya, and J. Plusquellic, "A physical unclonable function defined using power distribution system equivalent resistance variations," in *2009 46th ACM/IEEE Design Automation Conference*, pp. 676–681, 2009.

[6] B. Park, D. Forte, M. M. Tehranipoor, and N. Maghari, "A metal-via resistance based physically unclonable function with backend incremental adc," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 68, no. 11, pp. 4700–4709, 2021.

[7] M.-Y. Wu, T.-H. Yang, L.-C. Chen, C.-C. Lin, H.-C. Hu, F.-Y. Su, C.-M. Wang, J. P.-H. Huang, H.-M. Chen, C. C.-H. Lu, *et al.*, "A puf scheme using competing oxide rupture with bit error rate approaching zero," in *2018 IEEE International Solid-State Circuits Conference-(ISSCC)*, pp. 130–132, IEEE, 2018.

[8] A. Alvarez, W. Zhao, and M. Alioto, "14.3 15fj/b static physically unclonable functions for secure chip identification with¡ 2% native bit instability and 140× inter/intra puf hamming distance separation in 65nm," in *2015 IEEE International Solid-State Circuits Conference-(ISSCC) Digest of Technical Papers*, pp. 1–3, IEEE, 2015.

[9] J. Li and M. Seok, "Ultra-compact and robust physically unclonable function based on voltage-compensated proportional-to-absolute-temperature voltage generators," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 9, pp. 2192–2202, 2016.

[10] M. Danesh, A. B. Venkatasubramaniyan, G. Kapoor, N. Ramesh, S. Sadasivuni, S. T. Chandrasekaran, and A. Sanyal, "Unified analog puf and trng based on current-steering dac and vco," *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, vol. 28, no. 11, pp. 2280–2289, 2020.

[11] D. Jeon, J. H. Baek, Y.-D. Kim, J. Lee, D. K. Kim, and B.-D. Choi, "A physical unclonable function with bit error rate $< 2.3 \times 10^{-8}$ based on contact formation probability without error correction code," *IEEE Journal of Solid-State Circuits*, vol. 55, no. 3, pp. 805–816, 2019.

[12] S. Chowdhury, R. Acharya, W. Boullion, A. Felder, M. Howard, J. Di, and D. Forte, "A weak asynchronous reset (ares) puf using start-up characteristics of null conventional logic gates," in *2020 IEEE International Test Conference (ITC)*, pp. 1–10, IEEE, 2020.

[13] A. Maiti, V. Gunreddy, and P. Schaumont, "A systematic method to evaluate and compare the performance of physical unclonable functions," in *Embedded systems design with FPGAs*, pp. 245–267, Springer, 2013.

[14] M. Cortez, A. Dargar, S. Hamdioui, and G.-J. Schrijen, "Modeling sram start-up behavior for physical unclonable functions," in *2012 IEEE International Symposium on Defect and Fault Tolerance in VLSI and Nanotechnology Systems (DFT)*, pp. 1–6, IEEE, 2012.

[15] K. Deb, A. Pratap, S. Agarwal, and T. Meyarivan, "A fast and elitist multiobjective genetic algorithm: Nsga-ii," *IEEE transactions on evolutionary computation*, vol. 6, no. 2, pp. 182–197, 2002.

[16] K. M. Fant and S. A. Brandt, "Null convention logic: A complete and consistent logic for asynchronous digital circuit synthesis," in *Proceedings of International Conference on Application Specific Systems, Architectures and Processors: ASAP '96*, 1996.

[17] V. Singh, "A note on determination of oscillation startup condition," *Analog Integrated Circuits and Signal Processing*, vol. 48, no. 3, pp. 251–255, 2006.

[18] S. N. Sivanandam and S. N. Deepa, *Introduction to Genetic Algorithms*. Springer Publishing Company, Incorporated, 1st ed., 2007.

[19] K. Deb, "Multi-objective optimization," in *Search methodologies*, pp. 403–449, Springer, 2014.

[20] J. Blank and K. Deb, "pymoo: Multi-objective optimization in python," *IEEE Access*, vol. 8, pp. 89497–89509, 2020.